

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KỸ THUẬT MÁY TÍNH



BÁO CÁO BÀI THỰC HÀNH LAB02

HỌ VÀ TÊN: NGUYỄN GIA BẢO NGỌC – 21520366
NGUYỄN QUỐC TRƯỜNG AN – 21521810
NGUYỄN ĐỨC LƯU – 21522314

LỚP: CE224.O12.1

GIẢNG VIÊN HƯỚNG DẪN:
CHUNG QUANG KHÁNH

TP. HỒ CHÍ MINH – Tháng 10 năm 2023

MỤC LỤC

1. CHUẨN BỊ	1
1.1 Phần cứng	1
1.2 Phần mềm	1
1.3 Kiến thức.....	1
2. HƯỚNG DẪN	1
2.1 Làm việc với SDRAM	1
2.2 Làm việc với LCD	3
3. BÀI TẬP	4
Bài tập 1: Tìm hiểu và trình bày về gyroscope trên KIT STM32F4 Discovery?	4
Bài tập 2: Đọc dữ liệu từ gyroscope và hiển thị ra màn hình LCD?	5

LAB 2: LÀM VIỆC VỚI SDRAM VÀ LCD TRÊN STM32F4 DISCOVERY

1. CHUẨN BỊ

1.1 Phần cứng

- KIT STM32F4 Discovery for STM32F429 MCU.

1.2 Phần mềm

- STM32CubeIDE: sử dụng để lập trình, build, nạp và debug code.

1.3 Kiến thức

- Biết cách tạo project và cấu hình project sử dụng STM32CubeIDE.

2. HƯỚNG DẪN

2.1 Làm việc với SDRAM

Các bước đã thực hiện theo file hướng dẫn:

- Tạo project và thực hiện cấu hình xung clock chạy ở tốc độ cao nhất, sau lưu file IOC và sinh code.

- Thực hiện liên kết các gói thư viện BSP và HAL.

- Sau khi thực hiện liên kết và khai báo các thư viện, thực hiện chương trình kiểm tra hoạt động của SDRAM như sau:

```
23 /* USER CODE BEGIN Includes */
24 #include "stm32f429i_discovery_sdram.h"
25 #include "stm32f429i_discovery_lcd.h"
26 /* USER CODE END Includes */

45 /* USER CODE BEGIN PV */
46 volatile uint32_t value;
47 /* USER CODE END PV */

89 /* USER CODE BEGIN 2 */
90 BSP_SDRAM_Init();
91
92 *((uint32_t*)SDRAM_DEVICE_ADDR)=0x12345678;
93 value=*((uint32_t*)SDRAM_DEVICE_ADDR);
```

Câu hỏi: Đọc file stm32f4_discovery_sdram.h và cho biết:

```
60 /**
61  * @brief FMC SDRAM Bank address
62  */
63 #define SDRAM_DEVICE_ADDR ((uint32_t)0xD0000000)
64 #define SDRAM_DEVICE_SIZE ((uint32_t)0x800000) /* SDRAM device size in Bytes */
65
```

- Địa chỉ của SDRAM nằm ở đâu trong bộ nhớ ?

Địa chỉ bắt đầu của SDRAM: 0xD0000000

- Kích thước của SDRAM là bao nhiêu?

Kích thước của SDRAM (tính theo byte): 0x800000

Câu hỏi: Sử dụng chế độ debug, xem bộ nhớ ở vị trí của SDRAM và chứng minh đã gán giá trị 0x12345678 thành công vào SDRAM?

- Sử dụng chế độ debug và dùng chức năng monitor để giám sát vùng nhớ 0xD0000000 (vùng nhớ bắt đầu của SDRAM) để quan sát và nhận đã gán giá trị 0x12345678 thành công.

- Vì giá trị sẽ được nạp vào theo byte nên 4 bytes có địa chỉ bắt đầu là 0xD0000000 lần lượt được ghi giá trị 0x78, 0x56, 0x34, 0x12.

```
87 /* Initialize all configured peripherals */
88 MX_GPIO_Init();
89 /* USER CODE BEGIN 2 */
90 BSP_SDRAM_Init();
91
92 *((uint32_t*)SDRAM_DEVICE_ADDR)=0x12345678;
93 value=*((uint32_t*)SDRAM_DEVICE_ADDR);
94
95 BSP_LCD_Init();//init LCD
96 //set the layer buffer address into SDRAM
97 BSP_LCD_LayerDefaultInit(1, SDRAM_DEVICE_ADDR);
98 BSP_LCD_SelectLayer(1);//select on which layer we write
99 BSP_LCD_DisplayOn();//turn on LCD
100 BSP_LCD_Clear(LCD_COLOR_BLUE);//clear the LCD on blue color
```

Monitors				
0xD0000000 : 0xD0000000 <Hex> × New Renderings...				
0xD0000000	Address	0 - 3	4 - 7	8 - B
				C - F
	D0000000	78563412	FF0000BF	FF0000BF
	D0000010	FF0000FF	FF0000FF	BF0000FF
	D0000020	FF0000FF	FF8000FF	FF0000FB

2.2 Làm việc với LCD

Phần này thực hiện lấy dữ liệu từ SDRAM và hiển thị in lên màn hình LCD.

Thực hiện các bước theo hướng dẫn:

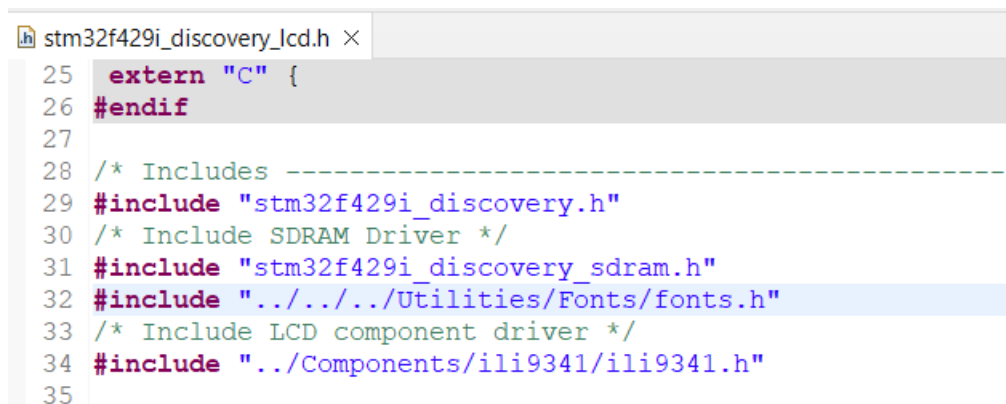
- Thực hiện liên kết các thư viện LCD và một số file định nghĩa Fonts chữ.
- Sau khi thực hiện liên kết và khai báo các thư viện, thực hiện chương trình kiểm tra hoạt động của LCD như sau:

```
23 /* USER CODE BEGIN Includes */
24 #include "stm32f429i_discovery_sdram.h"
25 #include "stm32f429i_discovery_lcd.h"
26 /* USER CODE END Includes */

95 BSP_LCD_Init(); //init LCD
96 //set the layer buffer address into SDRAM
97 BSP_LCD_LayerDefaultInit(1, SDRAM_DEVICE_ADDR);
98 BSP_LCD_SelectLayer(1); //select on which layer we write
99 BSP_LCD_DisplayOn(); //turn on LCD
100 BSP_LCD_Clear(LCD_COLOR_BLUE); //clear the LCD on blue color
101 BSP_LCD_SetBackColor(LCD_COLOR_BLUE); //set text background color
102 BSP_LCD_SetTextColor(LCD_COLOR_WHITE); //set text color
103 //write text
104 BSP_LCD_DisplayStringAtLine(2, "Cube STM32"); BSP_LCD_DisplayStringAtLine(3, "BSP");
105 BSP_LCD_DisplayStringAtLine(4, "LCD DEMO");
106 /* USER CODE END 2 */
```

Câu hỏi: Ta có thể thấy để sử dụng được module LCD ta sẽ cần không chỉ những thư viện xử lý mà còn cần một file định nghĩa font chữ. Dựa vào hướng dẫn hiển thị text trên LCD, hãy giải thích cách chương trình sử dụng file định nghĩa font này để hiển thị chữ trên màn hình LCD?

- File stm32f429i_discovery_lcd.h sẽ include các file header .h của thư viện định dạng Fonts chữ.



```
stm32f429i_discovery_lcd.h x
25 extern "C" {
26 #endif
27
28 /* Includes -----
29 #include "stm32f429i_discovery.h"
30 /* Include SDRAM Driver */
31 #include "stm32f429i_discovery_sdram.h"
32 #include "../Utilities/Fonts/fonts.h"
33 /* Include LCD component driver */
34 #include "../Components/ili9341/ili9341.h"
35
```

- Sau đó các file stm32f429i_discovery_lcd.c sẽ include file stm32f429i_discovery_lcd.h và các file định nghĩa hàm .c của thư viện định dạng Fonts chữ và sử dụng nó.

```
stm32f429i_discovery_lcd.c ×
55 -----
56
57 /* Includes -----
58 #include "stm32f429i_discovery_lcd.h"
59 #include "../Utilities/Fonts/fonts.h"
60 #include "../Utilities/Fonts/font24.c"
61 #include "../Utilities/Fonts/font20.c"
62 #include "../Utilities/Fonts/font16.c"
63 #include "../Utilities/Fonts/font12.c"
64 #include "../Utilities/Fonts/font8.c"
65
```

3. BÀI TẬP

Bài tập 1: Tìm hiểu và trình bày về gyroscope trên KIT STM32F4 Discovery?

Gyroscope sensor hay tên tiếng việt là cảm biến con quay hồi chuyển là một cảm biến dùng để đo tốc độ góc hay tốc độ quay của một đối tượng. Gyroscope hoạt động dựa trên nguyên tắc bảo toàn mô men động lượng, cảm biến gyroscope đo sự thay đổi trong chuyển động của đối tượng và chuyển nó thành tín hiệu điện để kiểm soát và theo dõi các phương, góc quay.

Gyroscope sensor được ứng dụng rộng rãi trong các thiết bị có liên quan đến đo đạc và theo dõi sự quay của đối tượng, chẳng hạn như các hệ thống nhúng bên trong ô tô, robot hay các thiết bị di động.

- Trong thiết bị di động, chúng được sử dụng để phát hiện hướng xoay của các thiết bị để thay đổi chế độ xoay màn hình hoặc phục vụ cho một số ứng dụng di động.
- Trong ô tô, các cảm biến được dùng để kiểm soát ổ đĩa tự động và cải thiện tính ổn định và an toàn.
- Trong ngành robotics, chúng được áp dụng để robot xác định vị trí và hướng di chuyển.

Trên KIT STM32F4 Discovery module Gyroscope sensor gồm các đặc điểm kỹ thuật:

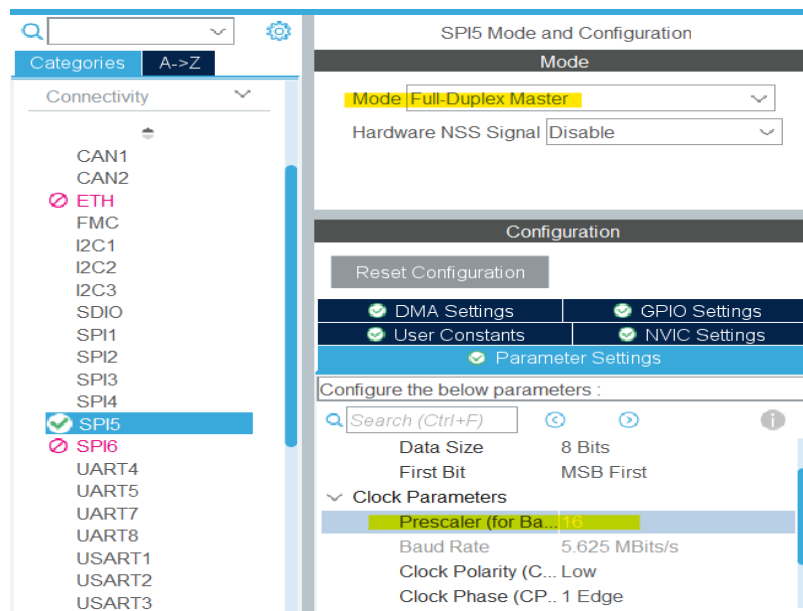
- Tên: Gyroscope MEMS (ST-MEMS I3G4250D)
- Là cảm biến tốc độ góc ba trục, công suất thấp
- Bao gồm một sensor và một IC interface
- Giao tiếp và truyền dữ liệu ra ngoài bằng I²C/SPI serial interface

- Có thang đo đầy đủ $\pm 245/\pm 500/\pm 2000$ dps và có khả năng đo tốc độ với băng thông do người dùng lựa chọn
- STM32F429ZIT6 điều khiển cảm biến chuyển động này thông qua SPI interface
- Hoạt động ở 2.4V – 3.6V
- Hoạt động nhiệt độ từ -40 độ C tới +85 độ C
- 16-bit tốc độ output và 8-bit nhiệt độ output

Bài tập 2: Đọc dữ liệu từ gyroscope và hiển thị ra màn hình LCD?

Ta thực hiện đọc dữ liệu từ gyroscope theo các bước sau:

- Bật Enable SPI5 để thực hiện nhận data từ Sensor



- Include thư viện **stm32f429i_discovery_gyroscope.h**

```

24 #include "stm32f429i_discovery_sdram.h"
25 #include "stm32f429i_discovery_lcd.h"
26 #include "stm32f429i_discovery_gyroscope.h"
27 /* USER CODE END Includes */

```

- Gọi hàm **BSP_GYRO_GetXYZ(float *ptr);** để lấy địa chỉ trả về của 3 tọa độ X, Y, Z

```

127 BSP_GYRO_GetXYZ(xyz_data_float_ptr);

```

- Chuyển data kiểu float thành string để hiển thị ra LCD

```
129     sprintf(str_x, "%.3f", xyz_data_float_ptr[0]);
130     BSP_LCD_DisplayStringAtLine(7, str_x);
131
132     sprintf(str_y, "%.3f", xyz_data_float_ptr[1]);
133     BSP_LCD_DisplayStringAtLine(9, str_y);
134
135     sprintf(str_z, "%.3f", xyz_data_float_ptr[2]);
136     BSP_LCD_DisplayStringAtLine(11, str_z);
```

❖ Kết Quả:

